

## I. Ejercicios programación en C

### Ejercicio 1

Escribir un programa para calcular el volumen de un cilindro:

- Definir variables para el radio de la base y la altura del cilindro.
- Leer los datos desde el teclado.
- Calcular el volumen.
- Mostrar el resultado en pantalla.

### Ejercicio 2

Escribir un programa que calcule la fuerza de atracción entre dos masas (*Ley de gravitación de Newton*: cualquier partícula de masa en el universo atrae a otra con una fuerza directamente proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia entre ellas). La fórmula es:

$$F = G \frac{m_1 \cdot m_2}{r^2}$$

Donde:

- $F$  es la fuerza de atracción entre las masas,
- $G$  es la constante gravitacional ( $6.674 \times 10^{-11} \text{ Nm}^2/\text{kg}^2$ ),
- $m_1$  y  $m_2$  son las dos masas, y
- $r$  es la distancia entre los centros de las masas.

El programa lee los datos desde el teclado. Utilizamos unidades del Sistema Internacional: fuerza en newton (N), masa en kilogramos (kg) y distancia en metros (m).

Comprobar el resultado calculando la fuerza con la que La Tierra ( $M=5,972 \times 10^{24} \text{ kg}$ ) atrae a una masa  $m = 1 \text{ kg}$ . El radio medio de La Tierra es 6371 km.

### Ejercicio 3

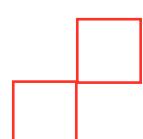
Escribir un programa que lee dos números, A y B, los ordena (copia en A el más pequeño) e imprime la lista ordenada.

### Ejercicio 4

Escribir un programa que lee tres números (a, b, c) y muestra el mayor (el menor).

### Ejercicio 5

Escribir un programa que lea un número correspondiente a un año e indique si el año es bisiesto o no. Un año es bisiesto si es divisible por 4 y no divisible por 100, o es divisible por 400. Ejemplo: 1900 no fue bisiesto, pero el año 2000 si lo fue.



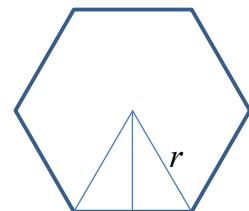
### Ejercicio 6

Escribir un programa que lea un número (1 para enero, 2 para febrero, ..., 12 para diciembre) y un año e indique el número de días de ese mes. Hay que tener en cuenta si el año fue bisiesto o no.

### Ejercicio 7

Escribir un programa que calcule el área de un hexágono regular:

$$A = 3r \sqrt{r^2 - \left(\frac{r}{2}\right)^2}$$



El radio,  $r$ , se lee desde el teclado.

Comprobar que el radio es positivo.

Utilizar la función `sqrt()` para calcular la raíz cuadrada, que está en la biblioteca **math.h**.

### Ejercicio 8

Escribir un programa que convierta una calificación numérica en su equivalente textual:

$0.0 \leq X < 5.0$	→ Suspenso
$5.0 \leq X < 7.0$	→ Aprobado
$7.0 \leq X < 9.0$	→ Notable
$9.0 \leq X < 10.0$	→ Sobresaliente
$X=10.0$	→ Matrícula de Honor

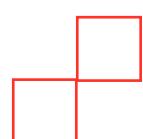
Comprobar que la nota numérica es válida, y si no, imprimir un mensaje de error y terminar el programa.

### Ejercicio 9

Escribir un programa que pida y lea los 3 coeficientes,  $a$ ,  $b$ ,  $c$ , de una ecuación de segundo grado ( $ax^2 + bx + c = 0$ ), y que calcule las soluciones.

Comprobar que el primer coeficiente ( $a$ ) es diferente de cero.

Las raíces pueden ser reales o complejas.



## II. Ejercicios programación en C (bucles)

### Ejercicio 1

Ejemplo de "switch":

```
*****  
* Una pequeña calculadora  
*****  
  
#include <stdio.h>  
int main ()  
{  
    float a=0.0, b=0.0;  
    char op;  
  
    printf ("Operacion: a op b \n");  
    scanf ("%f %c %f", &a, &op, &b);  
  
    switch(op) {  
        case '+': printf("= %g\n", a+b);  
        break;  
        case '-': printf("= %g\n", a-b);  
        break;  
        case '*': printf("= %g\n", a*b);  
        break;  
        case '/':  
            if(b==0)  
                printf("Division por cero\n");  
            else  
                printf("= %g\n", a/b);  
            break;  
        default : printf("Operador desconocido\n");  
    }  
    return 0;  
}
```

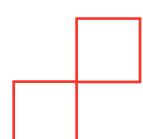
### Ejercicio 2

Escribir un programa que calcula la suma de los 10 primeros enteros ( $S = 1 + 2 + 3 + \dots + 10$ ).

### Ejercicio 3

Escribir un programa para imprimir una tabla de conversión de grados Fahrenheit a grados Celsius equivalentes (de 0°C a 100°C de 5 en 5).

$$g_F = \frac{9}{5}g_C + 32$$



### Ejercicio 4

Escribir un programa para imprimir una tabla de conversión de millas a kilómetros (1 milla = 1.609344 Km). Debe mostrar:

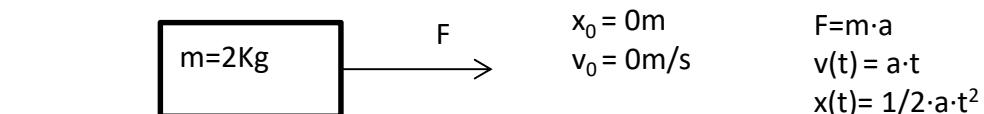
```

5 millas =     8.05 kilometros
10 millas =    16.09 kilometros
15 millas =    24.14 kilometros
.
.
.
100 millas =   160.93 kilometros

```

### Ejercicio 5

Un sistema se mueve sobre una superficie lisa sin rozamiento al aplicar una fuerza constante ( $F$ ), siendo la posición y la velocidad inicial nulas.



Escribir un programa que lee el valor de la fuerza aplicada  $F$  y muestra una tabla de tres columnas con el tiempo transcurrido (en incrementos de 1s), la velocidad y el espacio recorrido. Por ejemplo, si la fuerza es de 1N debe mostrar:

tiempo	velocidad	espacio
1	0.5	0.25
2	1.0	1.00
.		
.		
.		
10	5.0	25.00

### Ejercicio 6

Escribir un programa que imprime una tabla de las 10 primeras potencias de 2:

```

 $2^1 = 2$ 
 $2^2 = 4$ 
 $2^3 = 8$ 
...

```

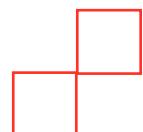
### Ejercicio 7

Escribir un programa que calcule el factorial de un número:

$$n! = 1 \cdot 2 \cdot 3 \cdots (n - 1) \cdot n, \text{ si } n > 0, \text{ y } 0! = 1$$

El programa lee  $n$  desde el teclado:

- Comprobar que  $n$  es válido, y si no, imprimir un mensaje de error que lo explique y terminar el programa.
- Comprobar que  $n$  es válido, y si no, pedir un nuevo valor.



### III. Ejercicios programación en C (funciones)

#### Ejercicio 1

a) Construya diagrama de flujo de un programa que calcula la suma de los ***n*** primeros enteros ( $S = 1 + 2 + 3 + \dots + n$ ). El programa debe realizar:

- Lea el número ***n*** desde el teclado. Compruebe que este número es positivo y si no es así, muestre un mensaje de error y termine el programa.
- Calcule la suma de forma iterativa.
- Imprima el valor calculado por pantalla.

b) Escribir el correspondiente programa en C que constará de las siguientes funciones:

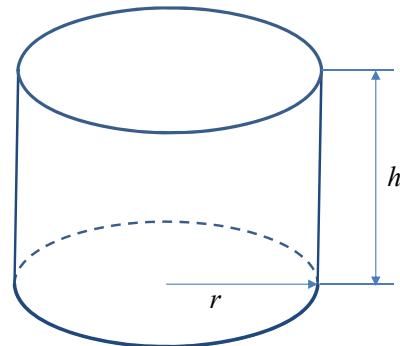
- Una función denominada ***SumaNumeros*** que reciba como argumento el número ***n*** y calcule y devuelva la suma ***S***.
- La función ***main*** en la que se pide, lee y comprueba el número ***n*** y llama a la función ***SumaNumeros*** para el cálculo y muestra por pantalla el resultado.

#### Ejercicio 2

Queremos calcular el volumen de un tanque cilíndrico y la cantidad de material necesario para construirlo:

a) Diseñar el organigrama de un programa que calcule el área y el volumen de un cilindro. Debe realizar lo siguiente:

- Leer el radio de la base y la altura del cilindro. Comprobar que son positivos y si no es así volver a pedirlos hasta que se introduzcan datos válidos.
- Calcular el área del cilindro y mostrar el resultado por pantalla.
- Calcular el volumen del cilindro y mostrar el resultado por pantalla.



b) Escribir el correspondiente programa en C que constará de las siguientes funciones:

- Una función denominada ***AreaCilindro*** que reciba como argumentos el radio ***r*** y la altura ***h*** de un cilindro y calcule y devuelva su área.
- Una función denominada ***VolumenCilindro*** que reciba como argumentos el radio ***r*** y la altura ***h*** de un cilindro y calcule y devuelva su volumen.
- La función ***main*** en la que se pide, lee y comprueba el radio y la altura y llama a estas funciones para calcular y mostrar por pantalla los resultados.

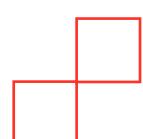
$$A = 2\pi \cdot r^2 + 2\pi \cdot r \cdot h, \quad V = \pi \cdot r^2 \cdot h$$

#### Ejercicio 3

Escribir una función llamada ***Bisiesto*** que reciba como argumento un número correspondiente a un año y devuelva un 1 si el año es bisiesto y un 0 si no lo es.

Un año es bisiesto si es divisible por 4 y no divisible por 100, o es divisible por 400.

*Ejemplo:* 1900 no fue bisiesto, pero el año 2000 si lo fue.



### Ejercicio 4

Escribir una función llamada **Metros2Yardas** que recibe como argumento un número real (el número de metros) y calcule y devuelva el número de yardas equivalentes (1 yarda = 0,9144 m).

### Ejercicio 5

Escribir una función llamada **Yardas2Metros** que recibe como argumento un número real (el número de yardas) y calcule y devuelva el número de metros equivalentes (1 yarda = 0,9144 m).

### Ejercicio 6

Escriba una función llamada **Hipotenusa** que reciba como argumentos los dos catetos de un triángulo rectángulo **c1** y **c2** y calcule y devuelva su hipotenusa. Utilice la función **sqrt( )** para el cálculo de la raíz cuadrada, que está en la biblioteca **math.h**.

### Ejercicio 7

- Escribir una función que calcule la potencia **n**-ésima de un entero **m**. Su prototipo es:  
`int potencia_int(int base, int exponente);`
- Escribir un programa que imprima una tabla con las 10 primeras potencias de 2.

### Ejercicio 8

- Escribir una función que calcule el factorial de un número **n**:  

$$n! = 1 \cdot 2 \cdot 3 \cdots (n - 1) \cdot n, \text{ si } n > 0, \text{ y } 0! = 1$$
- Escribir un programa que calcule el número combinatorio:

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

### Ejercicio 9

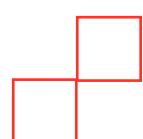
Escribir una función que convierta una calificación numérica en su equivalente con letras:

$0.0 \leq X < 5.0$	→ D
$5.0 \leq X < 7.0$	→ C
$7.0 \leq X < 9.0$	→ B
$9.0 \leq X \leq 10.0$	→ A

La función recibe como argumento un número real y devuelve su código equivalente.

### Ejercicio 10

Escribir una función que calcule el **n**-ésimo término de la serie de Fibonacci. La función recibe como argumento un número entero.



## IV. Ejercicios programación en C (vectores y matrices)

### Ejercicio 1

Diseñe el algoritmo (diagrama de flujo) y escriba un programa en C que realice las siguientes tareas:

- a) Lea un vector de 10 números reales desde el teclado.
- b) Busque el mayor de los elementos almacenados en el vector.
- c) Busque el menor de los elementos almacenados en el vector.
- d) Muestre por pantalla el mayor y el menor.
- e) Calcule la media de los 10 elementos.
- f) Muestre por pantalla el valor de la media

### Ejercicio 2

Escribir un programa que guarde en un vector los valores de los factoriales desde  $0!$  hasta  $12!$ . En el primer elemento del vector guardará  $0!$ , en el segundo  $1!$ , en el tercero  $2!$  y así sucesivamente.

$$n! = 1 \cdot 2 \cdot 3 \cdots (n - 1) \cdot n, \text{ si } n > 0, \text{ y } 0! = 1$$

### Ejercicio 3

Escribir un programa que guarde en un vector los primeros 20 números de la serie de Fibonacci.

### Ejercicio 4

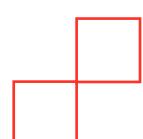
- a) Escribir una función llamada “**longcad**” que reciba como parámetro una cadena de caracteres y devuelva el número de caracteres que contiene.
- b) Escribir un programa que lea una cadena de caracteres desde el teclado y muestre en pantalla el número de caracteres introducidos.

### Ejercicio 5

Escribir un programa que lea una cadena de caracteres desde el teclado y una letra y cuente cuantas veces aparece esta letra en la cadena de caracteres.

### Ejercicio 6

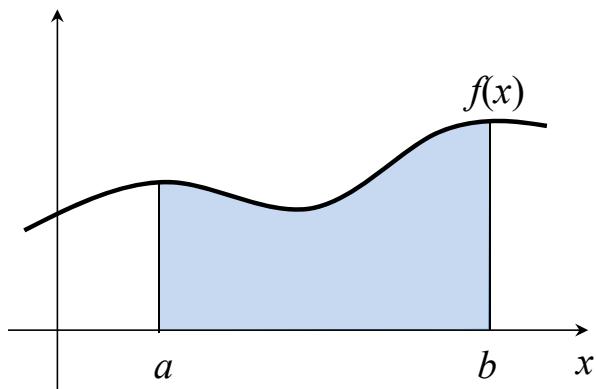
Escribir un programa que multiplique 2 matrices cuadradas de dimensión 3x3. Los elementos de cada matriz se introducen desde el teclado.



## V. Cálculo de la integral definida

### Introducción

La integral definida de una función  $f(x)$  puede escribirse  $\int_a^b f(x)dx$  donde  $a$  y  $b$  son los límites de integración. Gráficamente:



La integral definida puede calcularse como el área de la región en el plano  $xy$  (región sombreada), limitado por las líneas verticales  $x = a$ ,  $x = b$ , el eje  $x$ , y la curva de  $f$ .

### Cuestiones

El problema consiste en diseñar un algoritmo y escribir un programa en C para calcular una aproximación de la integral definida. Considerar que  $f(x)$  es una función continua. Los límites de integración,  $a$  y  $b$ , se leen desde el teclado.

Pasos:

1. Leer y analizar el problema. Pensar en las operaciones y en los datos (*entrada y salida*) que se necesitan para resolverlo. Proponer uno/varios métodos para resolverlo.
2. Diseñar el diagrama de flujo del programa que resuelve el problema.
3. Escribir un programa en C que resuelve el problema de acuerdo con el algoritmo diseñado. La función  $f(x)$  está codificada como una función de C que tiene como prototipo:

```
double fun_2(double x);
```

